
OpenCV project

1 Book Counting

After all the attempts i figured out that pre processing matters a lot.

The first preprocessing step is to remove Noise i.e text in our problem, for that i have used gaussian blurring,Normal averaging with kernel of 1s,some image cleaning functions like Morphology.I found that guassian with big sized kernels and MorphologyEx image cleaning giving good results.

By using **guassian blurr** i got this image



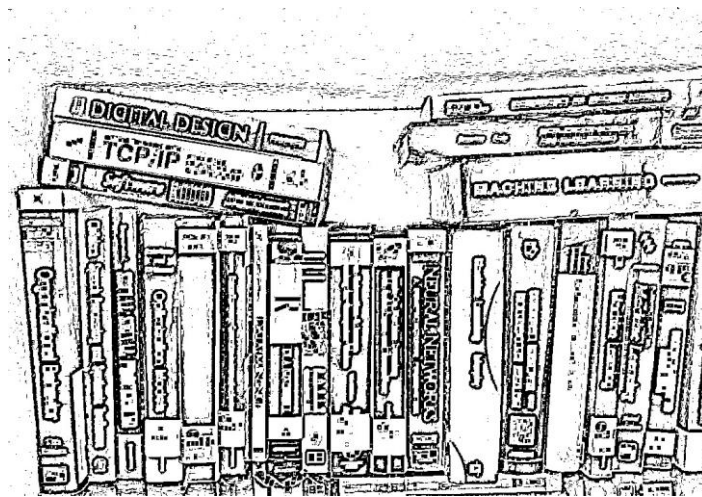
Figuur 1: Guassian Blurr By

using **Morphology erosion and dilation** i got this image



Figuur 2: Morphology

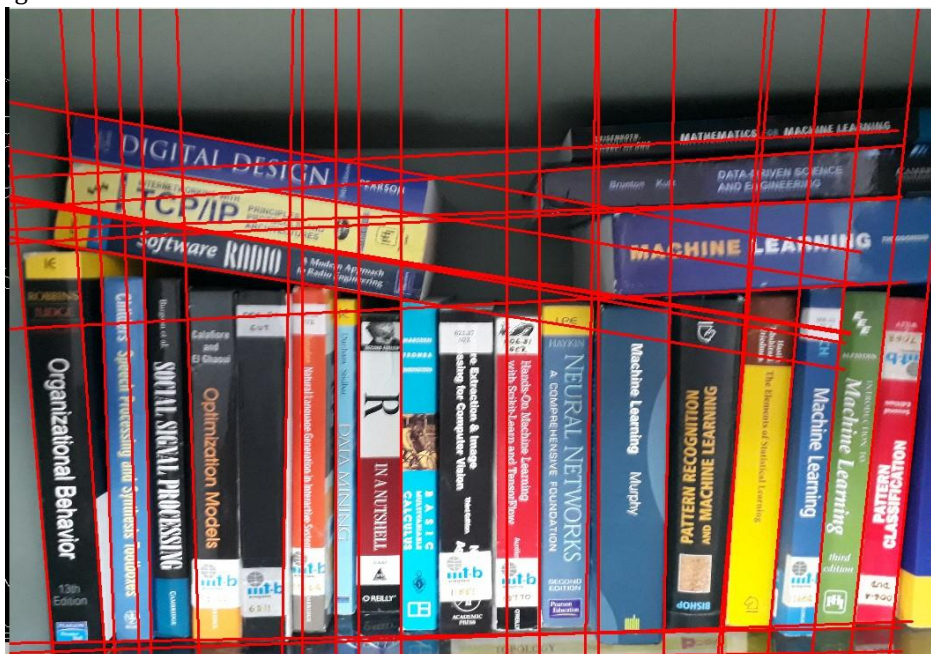
By adaptive guassian thresholding i can extract the information somewhat better.
By using **guassian threshold** i got this image



Figuur 3: Guassian Threshold

To detect the edges of the books i have used canny detection and for the threshold i have calculated from the median of the grey scale image. After finding the edges i have used some methods to count the books , they are:

1. **HoughLines:-** This technique is used to detect any lines on our image by HoughTransformation. So it basically transform every point to (r, θ) co ordinate system and it checks with the every possible line that passes through this point and it keeps on accumlate and finally it draws the line after certian number of points intersected in the (r, θ) system. By using **guassian blurr +canny + Houghlines** i got this image



Figuur 4: Houghlines

2. **Contours:-** It helps us to identify shapes in an image by joining all the points along the boundary of an image that are having the same intensity or color.

I got good results with houghlines than contours even though i blurred the text in the pre processing stage the text on the book is forming another shape, so the count is increasing when i use findContours method, whereas in houghlines i have changed the value of theta so it neglect the small lines of the text.

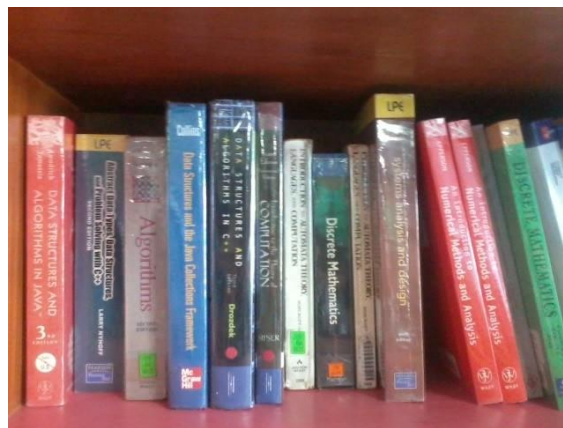
By using **MorphologyEx cleaning +canny + Contours** i got this image



Figuur 5: Contours

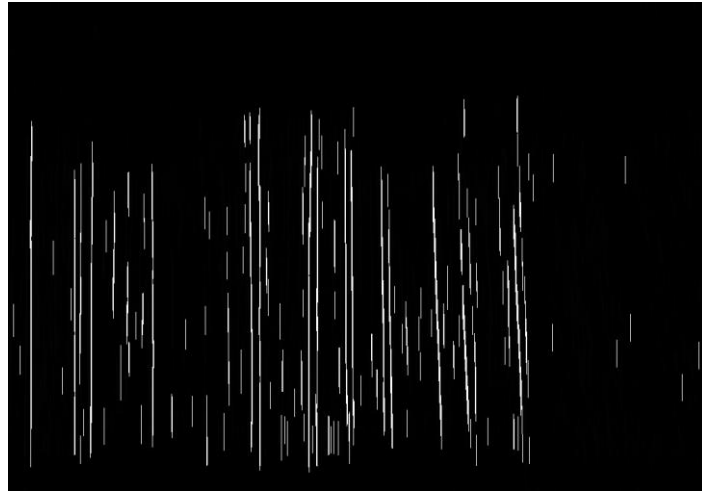
3. Using sobel and thresholding:

I applied the sobelx and sobely operators on the gaussian thresholded image to extract the horizontal and vertical lines. After getting sobelx output i.e vertical lines i applied vertical morphology erosion to only get the lines after that i removed the small connected components using the area of contours. I used different image that contains only vertically aligned books.



Figuur 6: Found in the internet

By using **thresholding+vertical morphology erosion** i got this image



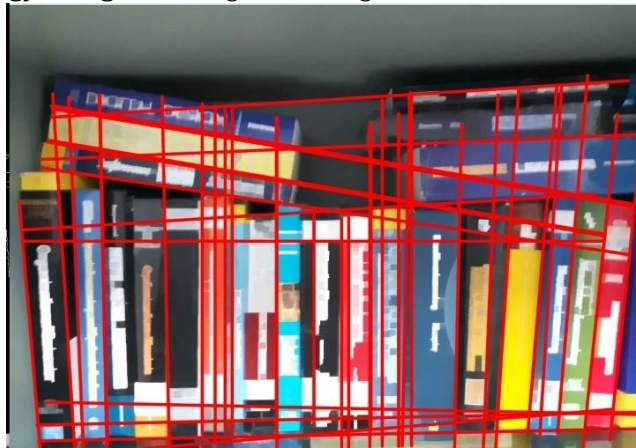
The same thing i have done with horizontal lines and counted the number of lines.

The problem i encountered by this method was ,i cannot remove the small lines and small connected components even though i removed them by finding area and height of the contours,many lines are getting removed.

4. **HoughLinesP:-**

This method basically takes random subset of points and find the lines it is same as houghlines but it takes only less number of points and it is faster than houghlines.

By using **Morphology+houghlinesP** i got this image



Figuur 7: HoughlinesP

I have tried many different methods to count the books but i found this 4 are giving better results than other methods and from the 4 i got close answer with houghlines.

2 **Face pixels**

To detect the face pixels from an image i used a mask to first find the skin cells. So firstly i removed some noise through median blurr and guassian blurr with some x sized kernal and then i converted our rgb space to hsv and YCbCr color space.After that i found the range of skin cells intensity from the internet then i created the mask with only skin cells and i applied the BitwiseAnd with the original image to get the image that contains only the skin cells.

Total process Lets blur the image using different ways.
By using **Gaussian blur**



By using **Median blur**



Now lets convert it into different spaces.
HSV space



Ycbr space



Skin Mask applied on Median blur + YCbr space



Skin Mask applied on Guassian blur + hsv space



As we can see that the bat of orange colour is also coming in the hsv space mask so i have taken Ycbr space range values of skin intensity.

Now lets apply the mask on our image to get the final results.

Using hsv mask



Using ycbcr mask



So from the above we can see that ycbcr mask is detecting better than hsv mask that we have created.

Some experiments I removed small contours from the mask with less area and i also tried to remove the contours that are attached to the big contour by retrieving the information from hierarchy parameter ,as we know that contour function first form the shape and go inside the shape to form another contours so i tried to remove the attached contours but i failed to do that and this information is stored in the hierarchy parameter.It doesnt work when there are many people in the image so it limits to single person and i also tried to remove noise by erosion and dilution.